

REMARKS

In the Office Action, the Examiner rejected Claims 31-34, which are all of the pending claims, under 35 U.S.C. §103 as being unpatentable over the prior art. Specifically, Claims 31, 33 and 34 were rejected as being unpatentable over U.S. Patent 6,078,744 (Wolczko) in view of U.S. Patent 6,367,012 (Atkinson, et al.), and Claim 32 was rejected as being unpatentable over Wolczko in view of U.S. Patent 6,317,872 (Gee, et al.).

Applicants herein ask that each of Claims 31-34 be amended to emphasize differences between the claims and the prior art. In particular, Applicants ask that these claims, which are all method claims, be amended to indicate expressly that these methods occur in a mixed static and dynamic environment.

The present invention, generally, relates to using a compiler to transform a computer program into executable code. The invention does this in two phases or steps. In one step, a compiler is used to perform a first set of tasks, and in another step, a virtual machine is used to perform a second set of tasks.

For example, Claim 31 is directed to a method in which statically compiled code may be securely executed by a virtual machine. In this method, the compiler and the virtual machine both perform tasks to ensure the integrity of the generated code.

Claim 32 is directed to a method for linking separately statically compiled code at run-time within a virtual machine by modifying the code. The compiler maintains certain symbolic entries, and the virtual machine used these symbolic entries in the generated code.

Claim 33 defines a method for updating statically generated code (C) at run-time when separately compiled code (S), which contained symbols referenced by C changes. In this method, the compiler generates certain code, and the virtual machine may recompile that code under certain conditions.

Claim 34 is directed to a method for maintaining full compliance with a language requiring dynamic compliance, while enabling the use of statically generated code for some byte code that depends on some byte code that may be separately compiled. The compiler performs certain security features on byte code, and the virtual machine uses those security features to determine if the byte code has changed.

What these claims all have in common, in addition to the use of a compiler and a virtual machine to perform two phases in the program transformation, is that they occur in a mixed static and dynamic environment. Moreover, the use of the compiler and the virtual machines, in the manner described in the claims, helps achieve this result – that is, operating in a mixed static and dynamic environment.

The prior art of record fails to disclose or teach the methods of Claims 31-34 in a mixed static and dynamic environment.

In particular, Wolczko, et al. discloses a procedure for improving the performance of a computer. In this procedure, compiled data that has been journaled is re-used.

The Examiner has recognized, however, that Wolczko fails to disclose or suggest a number of features described in Claims 31-34, and to address this deficiency of Wolczko, et al, the Examiner relies on Atkinson, et al. and Gee, et al.

Atkinson, et al. discloses a procedure to help ensure the integrity of executable files. With this procedure, a certification or signature may be incorporated in a file, and this certification or signature may be confirmed at the recipient computer.

Gee, et al, discloses a computer architecture for resolving symbolic references in code written in an object oriented programming language.

These references, however, whether they are considered individually or in combination, discloses or teach the methods of Claims 31-34 in a mixed static and dynamic environment. Furthermore, these references do not disclose the specific above-discussed distribution of tasks between a compiler and a virtual machine. It is this distribution, it may be noted, that facilitates performing these methods in that mixed static and dynamic environment.

Applicants herein request that Claims 31-34 be amended to emphasize the above-discussed aspect of this invention. Specifically, as mentioned above, Applicants ask that these claims be amended to indicate that the methods defined by the claims occur in a mixed static and dynamic environment.

This aspect of the invention provides a number of important advantages, and in particular, the invention combines the advantages of static and dynamic compilation. The invention effectively achieves the reduced performance overhead of dynamic compilers while still achieving the aggressiveness that can be achieved with static compilers.

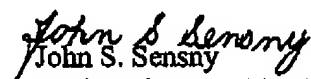
Because of the above-discussed differences between Claims 31-34 and the prior art, and because of the advantages associated with those differences, these Claims, as amended herein, patentably distinguish over the prior art and are allowable. Accordingly, the Examiner

is requested to enter this Amendment, and to reconsider and to withdraw the rejections of, and to allow, Claims 31-34.

The changes requested herein to Claims 31-34 only emphasize differences between the claims and the prior art. Moreover, it is noted that, even though the last Office Action was made final, this was the first Action in which Wolczko, et al, Atkinson, et al, and Gee, et al. were cited by the Examiner. Thus, this is the first opportunity Applicants have had to address these references. Consequently, it is believed that entry of this Amendment is appropriate, and such entry is respectfully requested.

For the reasons set forth above, the Examiner is requested to enter this Amendment, to reconsider and to withdraw the rejections of Claims 31-34 under 35 U.S.C §103, and to allow these Claims. If the Examiner believes that a telephone conference with Applicants' Attorneys would be advantageous to the disposition of this case, the Examiner is asked to telephone the undersigned.

Respectfully submitted,


John S. Sensny
Registration No. 28,757
Attorney for Applicants

Scully, Scott, Murphy & Presser
400 Garden City Plaza
Garden City, New York 11530
(516) 742-4343

JSS:jy